

 Yazılım proje yönetimi



Tekrar

# Etkin proje yönetimi

**İnsan Kaynakları:** Seçim, performans yönetimi, eğitim, kariyer geliştirme, organizasyon ve iş tasarımı, ekip kültürü geliştirme konuları üzerinde durulmalıdır.

**Problemler:** Bir proje planlanmadan önce amacı ve alanı belirlenmeli, alternatif çözümler düşünülmeli, teknik ve yönetim kısıtları tanımlanmalıdır.

**Süreç:** Yazılım geliştirme sürecinde gerçekleştirilen işlemler, esas işlemler ve bunların gerçekleşmesini destekleyen işlemlerden oluşur.

# YAZILIM ÖLÇÜMU

- Yazılım mühendisliği, yazılım ürününü oluşturmaya, mühendislik yaklaşımı uygulamakla ilgili olan teknikler toplamını tanımlamak için kullanılan terimdir. Mühendislik yaklaşımı, yönetme-maliyet hesabı, planlama, modelleme, analiz etme, tasarlama, gerçekleştirme ve bakım anlamındadır. Tüm bu adımlarda ölçme vardır.

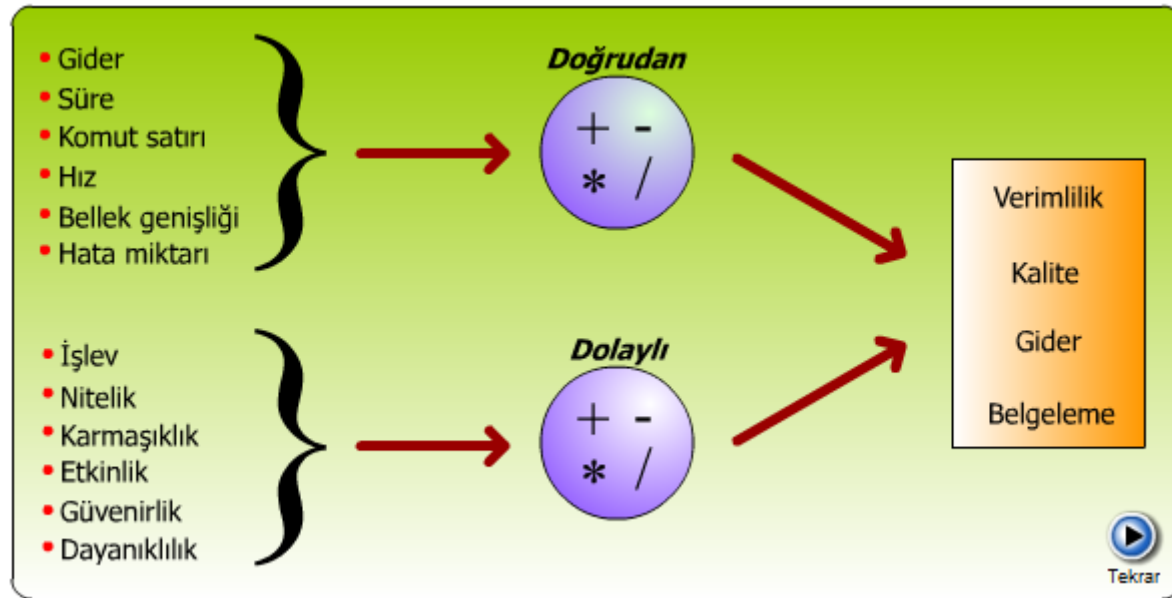
- Yazılım ürünü geliştirilen ölçülebilir hedefler belirlenememektedir. Örneğin; kullanım kolaylığı (user-friendly), güvenilir (reliable), bakım. Ancak bu özellikler için sayısal değerler verilmemektedir
- Yazılım projelerinin gerçek maliyetlerini oluşturan çeşitli elemanları ölçmek zordur.
- Ürettiğimiz ürünlerin kalitesini ölçemeyiz. Örneğin potansiyel kullanıcının belirli bir kullanım süresinde karşılaşacağı hata olasılığı veya farklı bir makine ortamında ne kadar sürede çalışabilir hale getirileceği gibi bir ürünün ne kadar güvenilir olabileceği söylenememektedir.
- Yeni gelişme teknolojisi veya aracının denenmesi sonucunda uygun olup olmadığı kesin ölçüm sonuçları ile değil anlatım yoluyla cevaplanmaktadır.
- Yazılım ürünü; ürün niteliğinin belirlenmesi, üretici verimliliğinin saptanması,

## YAZILIM KALİTESİ VE YAZILIM ÖLÇÜTLERİ

### YAZILIM ÖLÇÜMÜ ve ÖLÇÜM BİÇİMLERİ

- İki temel ölçüm biçimi vardır:
  - Doğrudan (Ör. gider, süre, komut satırı, hız, bellek genişliği, hata miktarı)
  - Dolaylı (Ör. işlev, nitelik, karmaşıklık, etkinlik, güvenilirlik, dayanıklılık)
- Ortak amaç; verimlilik, kalite, gider ve belgelemenin hesaplanmasıdır
- Doğrudan ölçülen niceliklerden tanımlanan bazı ölçütler:
  - $LOC = \text{Line of Code}$  (Kod satırı sayısı)
  - $KLOC = 1000 * LOC$  (K:Kilo)
  - $\text{Verimlilik} = KLOC / (\text{kişi} * \text{ay})$
  - $\text{Kalite} = \text{Hata} / KLOC$
  - $\text{Gider} = \text{Toplam gider} / KLOC$
  - $\text{Belgeleme} = \text{Belge sayfası} / KLOC$

# Yazılım Ölçüm Tipleri



|                                    | <i>Sayılar</i> | <i>Ağırlık faktörü</i> |             |                 | <i>FP</i> |
|------------------------------------|----------------|------------------------|-------------|-----------------|-----------|
|                                    |                | <i>Basit</i>           | <i>Orta</i> | <i>Karmaşık</i> |           |
| Kullanıcının yazılıma giriş sayısı |                | 3                      | 4           | 6               |           |
| Kullanıcının aldığı çıktı sayısı   |                | 4                      | 5           | 7               |           |
| Kullanıcının sorgulama sayısı      |                | 3                      | 4           | 6               |           |
| Kütük sayısı                       |                | 7                      | 10          | 15              |           |
| Dış arabirim sayısı                |                | 5                      | 7           | 10              |           |
| <i>Toplam</i>                      |                |                        |             |                 |           |

- **FP = sayısı \* ağırlık faktörü**
- **FP = Sayı toplamı  $(0.65 + 0.01 \sum F_i)$  ( $i = 1.... 14$ )**



# FP

- olarak hesaplanmaktadır. Yazılımın basit-orta-karmaşık oluşu tahmin yoluyla kestirilmektedir. Yazılımın 14 özelliğine göre (Tablo 2.2) "karmaşıklık düzeltme değeri" ( $\epsilon f_i$ ) bulunarak da
- **FP = Sayı toplamı ( $0.65 + 0.01 \epsilon F_i$ ) ( $i = 1.... 14$ )**
- fonksiyon noktası hesaplanabilmektedir.

| <i><b>Parametre</b></i>                                    | <i><b>Değer</b></i> |
|--|---------------------|
| Güvenli yedekleme ve geri yükleme gerekli mi ?             | 4                   |
| İletişim altyapısı gerekli mi ?                            | 2                   |
| Dağıtılmış işleme fonksiyonları var mı?                    | 4                   |
| Performans kritik mi?                                      | 3                   |
| Sistem yükü fazla mı?                                      | 5                   |
| Çevrimiçi veri girişi var mı?                              | 5                   |
| Çok ekranlı hareket girişleri var mı?                      | 1                   |
| Ana dosyalar çevrimiçi güncelleniyor mu?                   | 0                   |
| Giriş, çıkış ve sorgular karmaşık mı?                      | 1                   |
| İçsel işlemler karmaşık mı?                                | 1                   |
| Yeniden kullanılabilirlik var mı?                          | 3                   |
| Yükleme tasarıma dahil mi?                                 | 3                   |
| Farklı şirketlerde de çalışması söz konusu mu?             | 0                   |
| Uygulama kullanıcı tarafından kolayca değiştirilebilir mi? | 0                   |
| <b>TOPLAM</b>  | <b>32</b>           |

- **Verimlilik = FP/kişi \* ay**
- **Kalite = Hata/FP**
- **Gider = Toplam gider/FP**
- **Belgeleme = Belge sayfası/FP**

## YAZILIM KALİTESİ VE YAZILIM ÖLÇÜTLERİ

### YAZILIM ÖLÇÜMÜ: FONKSİYON NOKTASI YÖNTEMİ

- Kıyaslama ve eleştiriler:

| LOC ÖLÇEĞİ   | FP ÖLÇEĞİ                              |
|--|--|
| – Programlama diline bağımlı (kısmen)                                | + Programlama dilinden bağımsız        |
| – İyi tasarlanmış ama kısa yazılımları yeterince değerlendirememekte | – Sübjektiftir                         |
| – İşlemsel olmayan dillerdeki yazılımlara kolayca uyarlanamaz        | – Doğrudan fiziksel bir ölçüt değildir |
|  | – Veri toplaması güçtür                |

# YAZILIM PROJE MALİYET TAHMİN YÖNTEMLERİ

- Plânlama aşamasında yapılan proje tahmininin tutarlı ve güvenli olamayacağı saptanmıştır. Buna karşın yine de, eski bilgi ve deneyim sonuçlarına dayanarak, bir tahminde bulunmak gerekmektedir. Bu amaçla, çeşitli tahmin yöntemleri geliştirilmiştir:

bilirkişi takdiri - Delphi yöntemi –

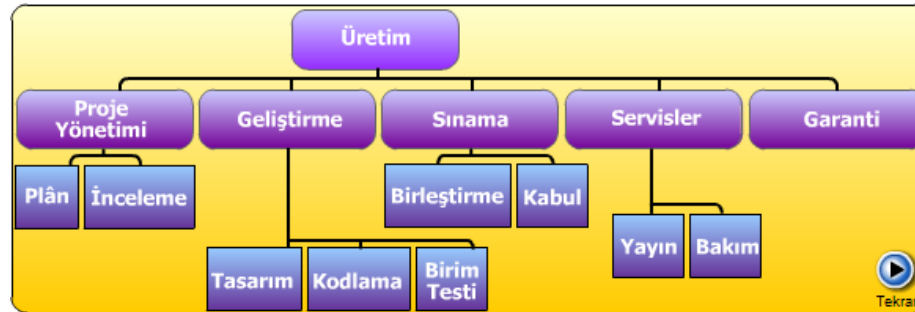
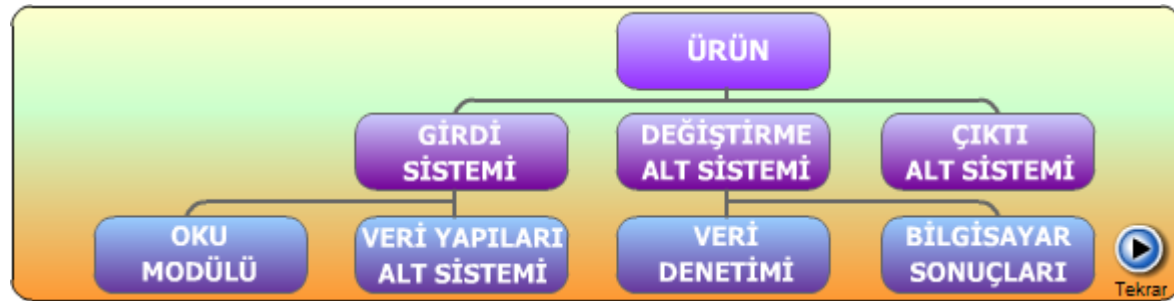
- analiz yöntemi - istatistik modeller

# DELPHI

## *DELPHI YÖNTEMİ AKIŞ DİYAGRAMI*



# ANALİZ YÖNTEMİ



# İSTATİSTİK MODELLER

- İstatistik modeller, doğrudan ölçülmeyen bir değeri (y), ölçülebilen bağımsız bir değişken (x) yardımı ile kestirmek amacı ile kurulan,
- $y = f(x)$
- istatistik fonksiyonlardır
- Yazılım proje tahmininde genellikle;
- $y = ax^b$
- 
- 
- Analiz yöntemi; ürün veya işlemi hiyerarşik olarak öğelerine ayırarak, en alt öğelerden başlayıp yukarı doğru her öğe için gider takdir etmektedir. Bu amaçla önce, ürüne ve/veya üretime dayalı analiz kartı düzenlenmektedir



# COCOMO

- Boehm; COCOMO (Constructive Cost Model) adını verdiği ilginç bir yazılım proje hesabı tahmin modeli geliştirmiştir. Model, hiyerarşik olarak üç basamak halinde uygulanmaktadır.
- **1. TEMEL COCOMO**  
Bin kod satırına (KLOC) dayanarak iş hacmi (H) kestirilmekte ve H serbest değişkenine bağlı olarak da proje süresi (T) hesaplanmaktadır.
- **$H = a KLOC^b$**
- 
- **$T = c H^d$**
- 
-

# COCOMO

- **2. ORTA COCOMO**

Temel COCOMO modeline ayrıca ürün-donanım-işgücü-proje özelliklerinden oluşan ve düzenlenen bir tablodan alınan bir “düzeltme faktörü” (EAF) de serbest değişken olarak eklenmektedir.

- **$H = a KLOC^b \cdot EAF$**

- 

-

# COCOMO modeli,

- COCOMO modeli, proje hesapları için geliştirilmiş en uygun yöntem olarak görülmektedir. Bu yöntemin uygulamasında izlenen işlemler:
- Üründeki bütün alt sistem ve modellerin tanımlanması,
- Her modülün büyüklüğünün tahmini ve bu tahminlere dayanarak, alt sistemler ile sistemin tamamının büyüklüğünün hesaplanması,
- Modül düzeyleri için, ilgili tablolarda ürün, donanım, personel, uygulanan teknoloji özelliklerine göre verilmiş olan düzeltme faktörlerinin (EAF) alınması,
- Her modül için, tablolardan alınan katsayılarla ve faktörlere göre istatistik fonksiyonlar oluşturularak iş hacmi (H) ve proje süresi (T) değerlerinin hesaplanması,
- Modül hesabı sonuçlarına göre, sistemin geliştirilmesi için gerekli toplam işgücü ve proje süresinin bulunması,
- Fayda/yarar değerlendirmesi için, duyarlık analizlerinin yapılması,
- Planlama ve analiz giderlerinin eklenmesi,
- Sonucun, Delphi yöntemi uygulanarak takdir edilen değer ile karşılaştırılması ve farklılığın giderilmesi
- aşamaları halinde verilmektedir.

# ÖRNEK

- Orta COCOMO basamağında da düzeltme faktörünün (EAF), duruma göre 0,90-1.40 arasında alınmasını öğütlemektedir. Örneğin, bir mikro işlemcide uzaktan haberleşme amacı ile kullanılan 10.000 satır kodlu gömülü (embedded) sistem yazılımında, düzeltme faktörü  $EAF = 1.17$  alınarak, geliştirilmesi için gerekli
- **İş Hacmi :  $H = 3.6 \cdot 101.20 \cdot 1.17 = 66.8$  programcı / ay**
- **Proje Süresi  $T = 2.5 \cdot 66.8 \cdot 0.32 = 9.6$  ay**

- **FP = Sayı toplamı  $(0.65 + 0.01 \varepsilon F_i)$  ( $i = 1.... 14$ )**

# PNR (Putnam-Norden-Rayleigh)

- istatistik tahmin modelidir.
- işgücü,  $H$ ; bin kod sayısı (KLOC), teknoloji katsayısı ( $C_k$ ) ve yıl olarak geliştirme sürecine ( $t$ ) bağlı;
- **$H = KLOC^3/(C_k^3t^4)$**
- istatistik fonksiyonu ile tahmin edilmektedir.  $C_k$  teknoloji katsayısı, düşük teknoloji ortamında 2000, iyi teknoloji ortamında 8000 ve otomatik yazılım araçlarının kullanıldığı ortamda 12000 olarak alınmaktadır. Eski verilere dayanarak da, en küçük kareler yöntemi ile kestirilebilmektedir. Bu modelde, işgücü ihtiyacının proje süreci basamaklarına, böylece de sürecin başından bu yana geçen zamana ( $t$ ) bağlı olduğu kabul edilmektedir.